Identify the equivalence class partitions and boundary values for the following requirements:
A teacher needs to enter into the system the student name and all the student's grades. All this information is entered in a single field, first the student name, followed by a comma " , ", then followed by the list of grades. A comma is used to separate the grades.
The student name is alphabetic characters 2 to 40 characters in length. Each grade is a value between 1 and 10, integer numbers only. The grades are entered in ascending order, so the smaller grades first. A maximum of 6 grades may be entered for each student.

Requirements breakdown:

*we have no separate name and surname, only name, without space, only alpha characters.
** There is an intermediate validation of the student name; the name input error message appears before the final submit, in case of invalid data, the teacher is not able to input coma& grades and then submit

1st Student Name	**2nd Second Grades**
Alpha characters only	Integer numbers
[2,40]	[1,10]
Followed by comma ","	Separated by comma ","
 Student name length	Ascending order
	Max 6 grades
	 Grades value
	 Number of ascending grades

Precond.	Func.	Equivalence class partitioning:	TD/BV	Type	Expected result
N/A	Student name length	[0,2)	0	I	User should not be able to register
			1	I	User should not be able to register
		[2,40]	2	V	User should be able to register
			40	V	User should be able to register
		(40, ∞)	41	I	User should not be able to register
		N/A	!@#$%^, 123456, ␣	I	User should not be able to register
Valid student name +','	Value grades	(-∞,1)	-1	I	User should not be able to register
			0	I	User should not be able to register
		[1,10]	1	V	User should be able to register
			10	V	User should be able to register
		(10, ∞)	11	I	User should not be able to register
		N/A	%; a	I	User should not be able to register
		N/A	1.3; 5.(9)	I	User should not be able to register
Ascendant input	No. of grades	[0,1),	0	I	User should not be able to register
		[1,6]	1	V	User should be able to register
			6	V	User should be able to register
		(6,∞)	7	I	User should not be able to register

Or….

TC Id.	TC Name	TD (Input)	Type	TC Id
1	[InputStudentName] Validate that teacher can only input alphabetic character for student name	!@#$%^, 123456, ␣	I	2
2	[InputStudentName] Validate that teacher cannot input non-alphabetic characters for student name	A	I	3
3	[InputStudentName] Validate that teacher cannot input student name with length 1	Al	V	1,4
4	[InputStudentName] Validate that teacher can only input alphabetic student name with length with length 2	Ionescualuipopescualuigeorgescualuipetru	V	1,5
5	[InputStudentName] Validate that teacher can only input alphabetic student name with length with length 40	Ionescualuipopescualuigeorgescualuipetrut	I	6
6	[InputStudentName] Validate that teacher cannot input student name with length 41 (or more)	1	I	7,2
7	[InputStudentName] Validate that teacher cannot input student grades without any student name (0 name)	Al1	I	8,2
Prec:	we have already the input of the valid student name	Al,-1	I	9
8	[InputStudentName&Grades] Validate that teacher cannot input grades without comma(“,”) between name and first grade	Al,0	I	10
Prec :	we have already the input of valid student name followed by comma	Al,1	V	1,11,17
9	[InputStudentName&Grades] Validate that teacher cannot input grade as a negative integer	Al,10	V	1,12,17
10	[InputStudentName&Grades] Validate that teacher cannot input 0 as a grade	Al,11	I	13
11	[InputStudentName&Grades] Validate that teacher can input 1 as a grade	Al + submit button	I	14
12	[InputStudentName&Grades] Validate that teacher can input 10 as a grade	Al,123456	I	15
13	[InputStudentName&Grades] Validate that teacher cannot input 11 (or more) as a grade	Al,10,2	I	16
14	[InputStudentName&Grades] Validate that teacher cannot submit only with name input, without any grades	Al,1,2,3,4,5,6	V	1, 11,18,19
15	[InputStudentName&Grades] Validate that teacher cannot input 2 up to 6 valid grades, which are not separated with “,”	Al,1,2,3,4,5,6,7	I	20
16	[InputStudentName&Grades] Validate that teacher cannot input 2 up to 6 valid grades, separated by comma “,”,which are not in ascendant order	Al,%; Al,a;	I	21
17	[InputStudentName&Grades] Validate that teacher is able to input one valid grade	Al,1.3; 5.(9)	I	22
18	[InputStudentName&Grades] Validate that teacher is able to input 6 valid grades, separated by comma “,”ordered ascendant	Al ␣ 1 ␣ 2 ␣ 3 ␣ 4 ␣ 5 ␣ 6; Al;1;2;3;4;5;6	I	23
19	[InputStudentName&Grades] Verify that a maximum of 6 grades can be entered			
20	[InputStudentName&Grades] Validate that teacher cannot input 7(or more) valid grades			

21	**[InputStudentName&Grades] Verify a grade that is not a numeric value cannot be added**
22	**[InputStudentName&Grades] Verify a grade that is not an integer cannot be added**
23	**[InputStudentName&Grades]Verify that data cannot be separated by other delimiters than comma**